AMENDMENTS TO THE SPECIFICATION

Please replace the paragraph on page 4, lines 1-10 with the following amended paragraph:

Another embodiment of the present invention may also be a computer system operative to detect whether an ATAPI drive is connected to an intelligent drive electronics channel within the computer. The computer system includes a processor coupled to a memory and a bus coupled to the processor that is capable of hosting at least one ATAPI drive via an intelligent drive electronics channel. The computer system also includes a BIOS program capable of being executed on the processor. When the BIOS program executing on the processor reads the cylinder high and cylinder low register destinations and detects that data read matches a predefined signature, the BIOS ~~programs return~~ program returns an indication that the ATAPI drive is connected to the intelligent drive electronics channel.

Please replace the paragraphs on page 6, lines 4-29 with the following amended paragraphs:

The system memory 22 includes read-only memory (ROM) 24 and random access memory (RAM) 25. A BIOS 26, containing the basic routines that help to transfer information between elements within the host computer 10, such as during start-up, is stored in ROM 24. The BIOS 26 also includes an IDE auto detection application 23 and [[a]] setup ~~menu~~ menus 20 utilized in the detection and initialization of the mass storage devices 27/29. Once the IDE auto detection application 23 assists the BIOS 26 in detecting whether a mass storage device 27/29 is present, the setup menu houses the initialized settings for the mass storage devices 27/29.

The computer 10 may also include an input/output interface controller 37 for receiving and processing input from a number of devices, including a keyboard or mouse (not shown). Similarly, an input/output interface controller 37 may provide output to a display screen, a printer, or other type of output device.

FIG. 2 illustrates the interface register destinations for the IDE drives according to an embodiment of the present invention. The IDE interface channels 32 each include an IDE interface controller 203 connected to the mass storage devices 27/29. The mass storage devices

27/29 each include drive controllers 212. Each drive controller 212 houses interface register destinations 202, 204, 206, and 208 for sending commands to the mass storage devices 27/29 or posting status from the devices 27/29. The interface register destinations are command block register destinations that are decoded by the IDE interface controllers 203. The command block register destinations include, among others, the device head 202, status 204, cylinder high 206, and cylinder low 208 register destinations. The contents of the register destinations are transferred to both the master 27 and the slave 29 devices and their embedded controllers 212. The host computer 10 discriminates between the two IDE devices 27 and 29 per channel 32 by using a bit in the device head register destination 202. Additional details regarding the bit structure of the device head register destination will be described below with respect to ~~FIG.3~~ FIG. 3.

Please replace the paragraph on page 6, line 30 through page 7, line 11 with the following amended paragraph:

Data is transferred in parallel either to or from host memory to the drive buffers 210 under the direction of the commands previously transferred from the host computer 10. The status register destination 204 contains the status of the device 27/29. The contents of this register destination are updated to reflect the current state of the device 27/29 and the progress of any command being executed in the device 27/29. After a reset condition and during POST, the IDE auto detection application 23 causes the BIOS 26 to read the status register destination 204 for a selected device 27/29. When the status register destination 204 indicates drive activity, the BIOS 26 receives an indication that the selected device or drive 27/29 is connected to the IDE interface channel 32. Additional details regarding the bit structure of the status register destination 204 will be described below with respect to ~~FIG.3~~ FIG. 3.

Please replace the paragraphs on page 7, lines 19-26 with the following amended paragraphs:

FIG. 3 illustrates the field bit structure of the interface register destinations of FIG. 2. The device select (DEV) bit, bit four (4) 302 of the device head register destination 202 discriminates between the master 27 and the slave 29 devices per IDE interface channel 32. For example, when the DEV bit is cleared to zero the device having the drive select value of zero is selected. When the DEV bit is set to one (1), the device having the drive select value of one is selected. Bits five and seven are obsolete bits, ignored by the drives 27/29. Other bits in the device/head register destination 202 are command dependent, denoted by '#'.

Please replace the paragraphs on page 7, line 27 through page 8, line 3 with the following amended paragraph:

Referring to the status register destination 204, when the combined data read from the status register destination 204 does not equal a zero (00h) hex value, standard IDE drive presence is detected. The busy (BSY) bit 303, bit 7, is set to one (1) to indicate that the selected device is busy, thereby indicating drive presence. The device ready (DRDY) bit 304, bit 6, is cleared to zero (0) by the device 27/29 during a reset condition when the device implements the packet command feature set. ATAPI devices implement the packet command feature set thus a zero hex value may be detected when an ATAPI device is attached.